

Bonus Assignment #3

(150 points)

Due Date: 5/6/19 by 11:59pm

Deliverable:

Post your homework as a SINGLE ZIP file on Blackboard with the name "Bonus_Assignment_3_YourLastName" that has the following:

1. Your IPYNB scrips; the entire python scripts that have your source code and the output
2. Your source code and output for every requirement numbered below. Create a cell for every requirement.
3. Your PDF files for your IPYNB script
4. Readme file how to install and run your assignment
5. Video recording of 10 minutes using screencast-o-matic (<https://screencast-o-matic.com/>) to demonstrate the run of your script and your output for the ipynb script

Important Notes:

- Please note that you can earn **either 0 or 150 points** in this Bonus assignment
- There is **NO partial credit** for this bonus assignment and there will be 0 credit for spaghetti code
- This assignment utilizes some of the work you chave done for Assignment #5

High-Level Description:

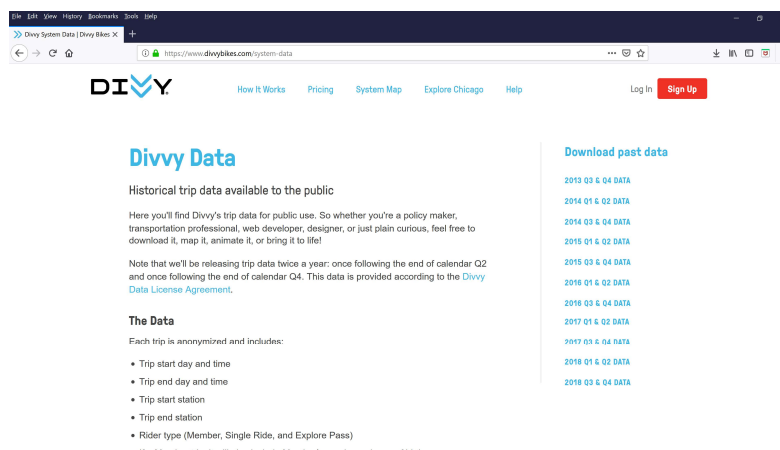
In this assignment you will analyze/visualize Divvy Bike Trips dataset and make forecast and detect riding patterns for the morning hours, evening hours, busiest day of the week, busiest month of the year, etc.

Please visit this URL (<https://www.divvybikes.com/system-data>) to learn more about Divvy Bikes dataset.

Chicago Divvy Dataset

Each trip is anonymized and includes:

- Trip start day and time
- Trip end day and time
- Trip start station
- Trip end station
- Rider type (Member, Single Ride, and Explore Pass)
- If a Member trip, it will also include Member's gender and year of birth



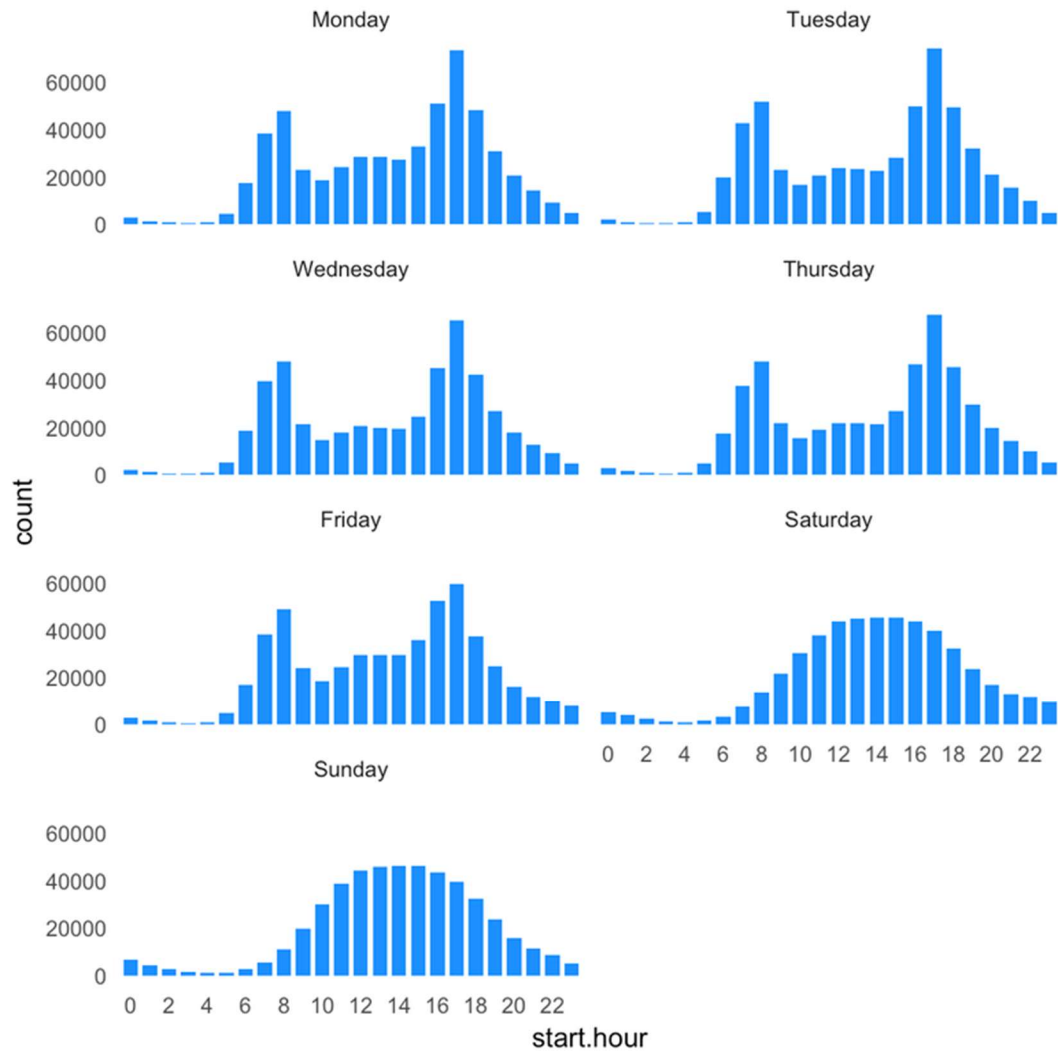
Requirements:

The intent of this assignment is to store Divvy Trips CSV dataset in ElasticSearch and plot and forecast time/day/week/month patterns for Divvy Bike Trips; that is, the seasonal patterns.

Create Python ipynb script based on the work you have done for Assignment #5 to achieve the following:

1. Store the given Divvy Trips CSV dataset on **ElasticSearch** server. You need to download ElasticSearch server from this URL:
(<https://www.elastic.co/downloads/elasticsearch>)
2. Read the data from ElasticSearch and load into a **DataFrame** object that you will call it divvy_trips.

3. Create **charts** that have the departures count by time-of-day similar to the following charts (notice you will create a bin for every hour of the day):



4. What is the **morning rush-hour** for Divvy Bike Riders? That is, maximum number of divvy trips during the morning rush-hour
5. What is the **evening rush-hour** for Divvy Bike Riders? That is, maximum number of divvy trips during the evening rush-hour
6. What is the **busiest day of week** for Divvy? That is, the day of the week that has the maximum number of trips
7. What is the **busiest week of the year** for Divvy? That is, the week of the year that has the maximum number of trips
8. What is the **busiest month of the year** for Divvy? That is, the month of the year that has the maximum number of trips
9. What is the **busiest from_station_name** for Divvy? That is, the from_station_name that has the maximum number of departures

9. Use Facebook/Prophet package (https://facebook.github.io/prophet/docs/quick_start.html) to forecast the following :
 1. Plot the rides forecast by calling the Prophet.plot method and passing in your forecast dataframe.
 2. Plot the rides forecast; use the Prophet.plot_components method. Plot the following 4 Trends: hourly, daily, weekly, yearly seasonality of the time series.
10. Re-implement the above 2 requirements (listed for Facebook prophet package) using TensorFlow Time Series (TFTS) <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/timeseries>
11. Re-implement the above 2 requirements (listed for Facebook prophet package) using StatsModel : <https://www.statsmodels.org/stable/index.html>
12. Discuss the results and metrics on how well Facebook/prophet, TensorFlow, and StatsModel performed to detect and forecast morning and evening rush hours, busiest day of the week, busiest week of the year, busiest month of the year.